

Appln No. 09/618,965

Amdt date December 24, 2003

Reply to Office action of October 9, 2003

REMARKS/ARGUMENTS

Claims 1-84 are currently pending in this application. Claims 1, 24, 25, 42, 43, 67, and 81-84 have been amended to place them in better condition for allowance. In view of the above amendments and following remarks, Applicants respectfully submit that the application is in condition for allowance. Applicants therefore, respectfully request reconsideration and allowance of the application.

This amendment is filed in response to the Supplemental Office action mailed October 9, 2002 which superceded the previous action mailed July 7, 2003. Applicants therefore assume that amendments to the claims made in Applicants September 30, 2003 response to the superceded action were not entered.

The Examiner rejected claims 1-84 under 35 U.S.C. 103(a) as being unpatentable over You et al. (U.S. Patent 5,815,653) in view of Cardoza et al. (U.S. Patent 5,630,049). The Examiner admits that You et al. does not explicitly teach monitoring at least a portion of data without disturbing operation. The Examiner alleges however that Cardoza et al. teaches a debugging system that uses status information on a portion of a target operating system to debug the target without disturbing operation thereof. The Examiner further alleges it would have been obvious to combine the teachings of Cardoza et al. with the debugging system of You et al. to provide a debugging system that monitors at least a portion of data without disturbing operation of a target construct. Applicants respectfully traverse this rejection.

Appln No. 09/618,965

Amdt date December 24, 2003

Reply to Office action of October 9, 2003

Independent claims 1 and 83 recite a method for interactive debugging comprised in part by "accessing data related to an operation of the target construct by a debug construct in real time; monitoring at least a portion of the accessed data without disturbing the operation of the target construct; and debugging the target construct using the monitored portion of the accessed data." Applicants respectfully submit that the cited references, alone or in combination do not disclose or suggest the recited limitations.

Rather, You et al. teaches that debuggers provide "information about the execution state of the running program as well as control of it. Program state includes program and data memory; hardware registers; program stacks; and operating system objects such as queues, synchronization objects, and program accounting information. Debuggers control programs with operations to start, stop, suspend, terminate, step over instructions, step into branches, step over statements, step through subroutine calls, stop at breakpoints, and stop at data watchpoints." (You et al., col. 2, lines 57-66). You et al., further discloses that "interactive program debuggers allow the user to inspect and control the execution of programs. They may also allow the user to modify the program's state and change the program's execution to behavior that is different from the semantics of the original source programs." (You et al., col. 6, lines 37-43).

Similarly, Cardoza et al. discloses a target computer system that may operate in one of two modes, polling mode, or interrupt-driven mode. Cardoza et al. further teaches that when "in polling mode, the target computer system is in a frozen or stopped state to enable, for example, examination of a programming variable that is used in operating system code being tested, and examination of register contents. Further, in

Appln No. 09/618,965

Amdt date December 24, 2003

Reply to Office action of October 9, 2003

polling mode, many of the target computer system's normal operations are suspended and the target computer system generally operates under the control of the remote debugger." (Cardoza et al., col. 10, lines 56-66, underlining added for emphasis only).

Thus, both You et al and Cardoza et al. disclose debugging systems that place the target computer system in a stopped state to enable the examination of program variables. However, neither You et al. or Cardoza et al. disclose or suggest monitoring at least a portion of the accessed data without disturbing the operation of the target construct and debugging the target construct using the monitored portion of the accessed data as recited in claims 1 and 83 of the present invention. Applicants therefore respectfully submit that claims 1 and 83 are novel and unobvious over the cited references and are therefore allowable. Applicants further submit that claims 2-23, that depend directly or indirectly from claim 1 are allowable as is claim 1 and for additional limitations recited therein.

The Examiner further admits that You et al. fails to disclose a method for multi-channel, multi-service debugging that includes "maintaining an isolated debugging environment for each of the plurality of running services" as recited in independent claims 24 and 84. The Examiner alleges however that Cardoza et al. discloses maintaining an insulated debugging environment for the computer resources. The Examiner further alleges it would have been obvious to combine the isolated environment of Cardoza et al. with the run time debugging environment of You et al. to provide the isolated debugging environment as recited in claims 24 and 84 of the present invention. Applicants respectfully traverse this rejection.

Appln No. 09/618,965

Amdt date December 24, 2003

Reply to Office action of October 9, 2003

As noted by the Examiner, Cardoza et al. discloses an "apparatus for software testing in an efficient, isolated, flexible, and controlled software testing environment which improves software engineering productivity, increases product quality, and reduces the use of computer resources and developer time." (Cardoza et al., col. 2, lines 35-41). The debugging system of Cardoza et al. includes "a computer network that comprises a host computer system and a target computer system. The software being tested resides in the target computer system. The host computer system includes a host operating system and a software debugger that controls testing of the software. The method comprising the steps of establishing, using the software debugger, a network connection between the host and the target computer system, transitioning the target computer system into a stopped state in which it awaits input from the host computer system..." (Cardoza et al., col. 2, lines 44-52).

Thus, the isolated environment of Carodza et al. simply refers to the isolation of the host computer system and the target computer system. Cardoza et al. does not however disclose or suggest maintaining an isolated debugging environment for each of the plurality of running services as recited in independent claims 24 and 84 of the present invention. Applicants therefore respectfully submit that claims 24 and 84 are novel and unobvious over the cited references and are therefore allowable. Applicants further submit that claims 25-41 that depend directly or indirectly from claim 24 are allowable as is claim 24 and for additional limitations recited therein.

Similarly, independent claim 42 recites an apparatus for interactive debugging comprised in part by "means for accessing data related to an operation of the target construct by a debug construct in real time; and means for monitoring at least a

Appln No. 09/618,965  
Amdt date December 24, 2003  
Reply to Office action of October 9, 2003

portion of the accessed data without disturbing the operation of the target construct." Applicants respectfully submit that the cited references do not disclose or suggest the recited limitations.

Rather, as argued above with respect to claim 1, both You et al and Cardoza et al. disclose debugging systems that place the target computer system in a stopped state to enable the examination of program variable. However, neither You et al. or Cardoza et al. disclose or suggest monitoring at least a portion of the accessed data without disturbing the operation of the target construct as recited in claim 42 of the present invention. Applicants therefore respectfully submit that claim 42 recites a novel and unobvious apparatus over the cited references and is therefore allowable.

Further, independent claim 43 recites an apparatus for multi-channel, multi-service debugging comprised in part by "means for maintaining an isolated debugging environment for each of the plurality services; and means for selecting a target construct for debugging from the plurality of running services." Applicants respectfully submit that the cited references do not disclose or suggest the recited limitations.

Rather, as argued above with respect to claim 24 the isolated environment of Carodza et al. simply refers to the isolation of the host computer system and the target computer system. Applicants therefore respectfully submit that claim 43 recites a novel and unobvious apparatus over the cited references and is therefore allowable.

Similarly, independent claim 44 recites an apparatus for interactive debugging comprised in part by "a debug construct configured to access data related to an operation of the target construct in real time and to monitor at least a portion of the data without disturbing the operation of the target construct."

Appln No. 09/618,965

Amdt date December 24, 2003

Reply to Office action of October 9, 2003

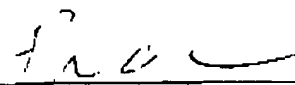
Further, claim 81 recites a system for interactive debugging comprised in part by at least one processor "configured to select the target construct for debugging, access the data in the memory in real time, and monitor at least a portion of the accessed data from the memory without disturbing the operation of the target construct to debug the target construct using the monitored portion of the accessed data." As argued above with respect to claim 1, applicants respectfully submit that the cited references, alone or in combination do not disclose or suggest the recited limitations. Applicants therefore respectfully submit that claim 81 is novel and unobvious over the cited references and is therefore allowable.

In addition, independent claim 82 recites a system for multi-channel, multi-service debugging comprised in part by "at least one processor coupled to the memory, the processor configured to maintain an isolated debugging environment for each of the plurality of running services and to provide a capability to view the information stored on the memory and to select a target construct for debugging from the information." As argued above with respect to claim 24, applicants respectfully submit that the cited references, alone or in combination do not disclose or suggest the recited limitations. Applicants therefore respectfully submit that claim 82 recites a novel and unobvious apparatus over the cited references and is therefore allowable.

Appln No. 09/618,965  
Amdt date December 24, 2003  
Reply to Office action of October 9, 2003

It is therefore respectfully submitted that pending claims 1-84 are in condition for allowance, and an early notice of allowance is respectfully requested.

Respectfully submitted,  
CHRISTIE, PARKER & HALE, LLP

By   
Peter A. Nichols  
Reg. No. 47,822  
626/795-9900

PAN/pan  
CAH PAS542469.1--12/23/03 3:12 PM



Creation date: 10-21-2004  
Indexing Officer: RDICKEY - RAYMOND DICKEY  
Team: OIPEBackFileIndexing  
Dossier: 09618965

Legal Date: 12-29-2003

No.	Doccode	Number of pages
1	A...	3
2	REM	1

Total number of pages: 4

Remarks:

Order of re-scan issued on .....